Subject: Re: Speed does matter
Posted by on Wed, 26 Aug 2015 08:41:31 GMT
View Forum Message <> Reply to Message

Le mardi 25 août 2015 23:35:11 UTC+2, Michael Galloy a écrit :
> On 8/25/15 7:03 AM, Kallisthène wrote:
>> Le mardi 25 août 2015 10:32:22 UTC+2, superchromix a écrit :
>>> Nice comparison. On my desktop, running IDL 8.5 and Matlab 2010b,
>>> I get the following results:
>>>
>>> IDL: % Time elapsed: 6.4400001 seconds.
>>>
>>> MATLAB 2010b: Elapsed time is 0.594112 seconds.
>>>
>>> Again, MATLAB for loops run faster by a factor of ~10.
>>
>> Great idea,
>>
>>
>> I ran also these codes, with bigger numbers, 1000 instead of 500 for
>> each dimension and with Matlab 2015a I got 6.3 secondes to compare
>> with 102 secondes on my IDL 8.5. My desktop runs 4 cores and has 12
>> Go RAM and it runs faster with Matlab by a factor of 16 ! Damn.
>>
>
> For what it's worth, I got about a 5x speedup of IDL over Python. I used
> 1000 elements for each dimension:
>
> $ python mg_loopspeed.py
> 222.15 secs
>
> Code is:
>

```
>    import numpy as np
>    import time
>
>    nx = 1000
>    ny = 1000
>    nz = 1000
>
>    arr = np.zeros((nx, ny, nz))
>
>    tic = time.clock()
>
>    for z in xrange(nz):
>        for y in xrange(ny):
>            for x in xrange(nx):
>                arr[x, y, z] = 1.0
```

>
> ```
>    toc = time.clock()
>    print '%0.2f secs' % (toc - tic)
> ```
>
> For comparison, my IDL time:
>
> ```
> $ idl -e ".run mg_loopspeed"
> IDL Version 8.5, Mac OS X (darwin x86_64 m64).
> (c) 2015, Exelis Visual Information Solutions, Inc., a subsidiary of
> Harris Corporation.
>
> % Compiled module: $MAIN$.
> % Compiled module: TIC.
>   Elapsed Time:  47.486635
> ```
>
> Mike
> --
> Michael Galloy
> www.michaelgalloy.com
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)

That's right, I got similar results on my Winpython 3.4. The reason seems to lie in the Just In Time compilation heavily used by Matlab (and Julia as well). To check it I used the "feature JIT off" undocumented Matlab command to see how it would fare.
And I got 48 secondes instead of the previous 6.3 secondes, an impressive change which puts Matlab un-JITed in the same league as IDL (102 secondes).
It is likely that's the reason.
To go further on this point with Python we'd need to check the efficiency of the Pypy Python interpreter and JIT compiler.

Seems to me that the work necessary to develop a JIT compiler for IDL is much heavier than compiling IDL against MKL.

Best

Best